实验报告

2154307 卢奕

一、问题描述与要求

理论和实验证明，一个两层的 ReLU 网络可以模拟任何函数。请自行定义一个函数，并使用基于 ReLU 的神经网络来拟合此函数。要求自行在函数上采样生成训练集和测试集，使用训练集来训练神经网络，使用测试集来验证拟合效果。

二、函数定义

$$y = 4(x+1)^3 + x^2$$

三、数据采集

- 在-3 到 3 区间内采集 1000 个样本点并进行打乱。
- 按照 6：4 的比例划分为训练集和测试集

```
# 采集数据
X = torch.unsqueeze(torch.linspace(-3, 3, 1000), dim=1)
Y = myfunction(X)
# 打乱数据
indices = torch.randperm(X.size(0))
X = X[indices]
Y = Y[indices]
# 划分训练集和测试集
trainX = X[:600]
trainY = Y[:600] + torch.normal(0, 0.1, size=(600, 1))
testX = X[600:]
testY = Y[600:]
```

四、模型描述

```
class myNet(nn.Module):
    def __init__(self):
        super(myNet, self).__init__()
        self.fc1 = nn.Linear(1, 500)
        self.fc2 = nn.Linear(500, 1)

    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

创建了一个两层的 relu 网络，网络有一个隐藏层，将输入特征映射到具有 500 个神经元的隐藏层后，使用 relu 激活函数引入非线性。

五、模型训练

1. 损失函数选择：均方误差（MSE）作为损失函数

2. 优化器选择：Adam 优化器

3. 超参数设置

（1）学习率：0.01

（2）训练周期：5000

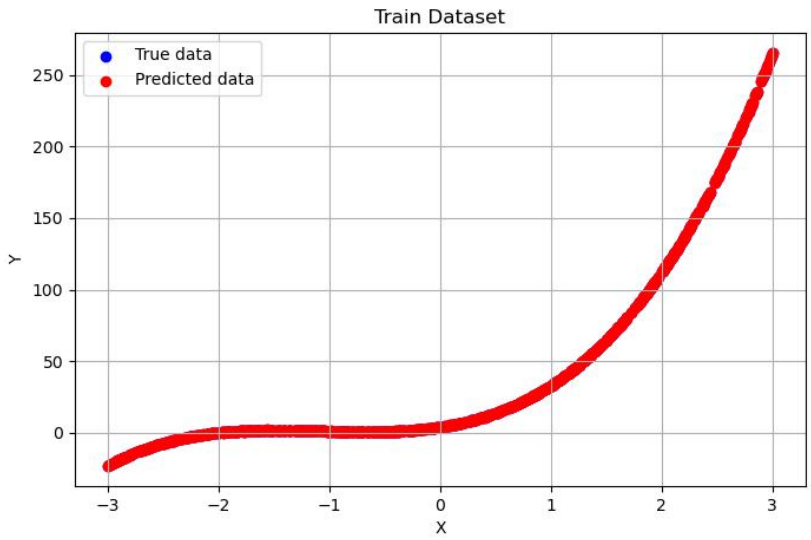
4. 在每个周期的结束时，记录训练集损失，并检查是否出现了更好的模型。最终选择具有最佳训练集损失的模型参数作为最终模型。

六、拟合效果

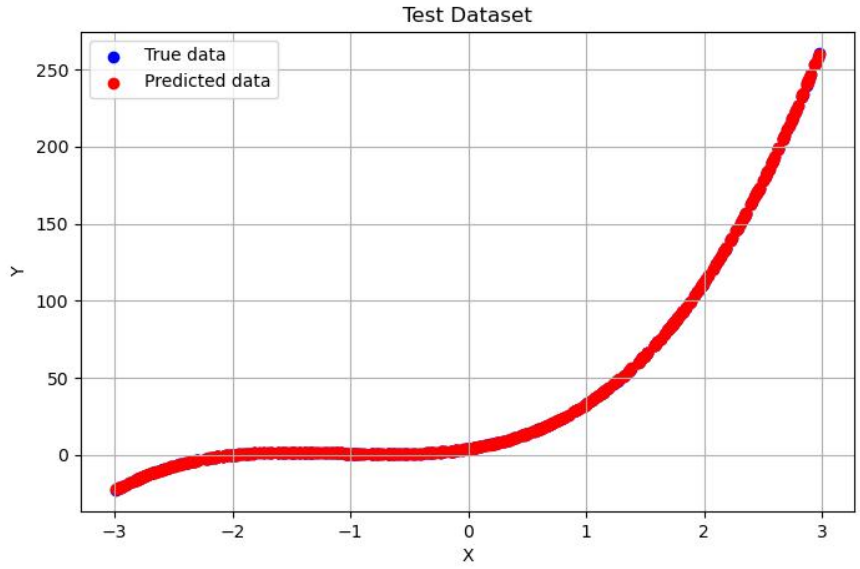
1. 经过 5000 轮迭代之后，模型在测试集上的 loss 为 0.0053

2. 拟合效果曲线

（1）训练集上:



（2）测试集上：


Test Dataset
True data
Predicted data
250
200
150
100
50
0
Y
−3
−2
−1
0
1
2
3
X